

Paavo Koivula

WLAN-liitännäyksikön ohjelmiston suunnittelu ja toteutus

Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2018



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tiivistelmä

Tekijä(t): Paavo Koivula

Työn nimi: WLAN-liitäntäyksikön ohjelmiston suunnittelu ja toteutus

Tutkintonimike: Insinööri (AMK)

Asiasanat: Wlan, Moduuli

Opinnäytetyön aiheena oli suunnitella ja toteuttaa ohjelmisto liitäntäyksikölle, joka mahdollistaisi langattoman tiedonsiirron laitteelle, johon liitäntäyksikkö kytkettäisiin. Liitäntäyksikön päätehtävänä oli siirtää kytketyltä laitteelta saatua dataa reitittimelle. Tiedonsiirtoon käytettiin WLAN-tekniikkaa. Työssä tavoitteena oli tuottaa toimiva ohjelmisto liitäntäyksikölle.

Työn tilaajana oli Devecto Oy, mutta itse työssä valmistunut ohjelmisto tuli Devecton asiakkaalle. Devecto Oy on tuotekehitystalo, jonka erikoisosaamista on älykkäiden koneiden ja laitteiden ohjelmistokehitys.

Työssä aluksi suunniteltiin ja toteutettiin ohjelmisto kehitysalustalle. Ohjelmiston tekemiseen käytettiin system workbench for STM32 ohjelmointiympäristöä. Työn aikana oli ongelmia kehitysalustan WLAN-moduulin kanssa. Lopputuloksena saatiin toimiva ohjelmisto kehitysalustalle.

Abstract

Author(s): Paavo Koivula

Title of the Publication: Software Design and Implementation for WLAN Expansion Board

Degree Title: Engineer

Keywords: Wlan, Module

The aim of this Bachelor's thesis was to design and implement software for the expansion board which would allow the host device to transfer data wirelessly. The expansion board's main task was transferring data from the host device to gateway. Data transferring was done using WLAN.

The thesis was commissioned by Devecto Oy but software was done for one of Devecto's clients. Devecto Oy is a software R&D company specializing in demanding software development for smart machines and devices.

Initially, the software was designed and implemented on the development platform. During the work there were problems with the development board's WLAN module. Working software was developed for the development board during this thesis process.

Alkusanat

Aloitin opinnäytetyöni tekemisen Kajaanissa alkuvuonna 2018. Työn aikana opin paljon uutta ohjelmiston suunnittelusta, ohjelmoinnista sekä virheen etsinnästä sulautettujen laitteiden ympäristössä.

Kajaanissa 9.4.2018

Paavo Koivula

Sisällys

Lyhenteet ja käsitteet	1
1 Johdanto	1
2 WLAN	2
2.1 IEEE-standardi	2
2.2 Kerrosmalli	3
2.2.1 Fyysinen kerros	4
2.2.2 Siirtokerros	4
2.3 WLAN verkon kanavointi	5
3 HART-protokolla	6
3.1 Viestien rakenne	6
3.2 Laitteiden yhteensopivuus	7
4 STMicroelectronics	8
4.1 Wi-Fi moduulit	8
4.1.1 SPWF01SA-moduuli	8
4.1.2 SPWF04SA-moduuli	9
4.2 Mikrokontrollerit	10
4.3 Kehitysalustat	10
4.3.1 NUCLEO-L476RG	10
4.3.2 X-NUCLEO-IDW01M1	11
5 Ohjelmiston vaatimusmäärittely	12
5.1 JSON formaatti	12
5.2 Sarjaportin toiminnan määrittelyn tarkennus	12
5.3 HART tuen määrittelyn tarkennus	13
5.4 WLAN määrittelyn tarkennus	13
5.5 Käyttöjärjestelmä	14
5.6 Ohjelmiston arkkitehtuuri	14
5.7 Käyttötapaukset	15
5.8 Ohjelmiston logiikka	16
6 Toteutus	17
6.1 Työkalut	18
6.2 Käytetyt väliohjelmistot ja HAL-kirjastot	18
6.3 Ohjelmistonkehitys ja ohjelmointi	19

6.3.1	WLAN-moduulin alustus ja käynnistys	20
6.3.2	WLAN-moduulin konfigurointi sekä verkkoon liittäminen	21
6.3.3	TCP-yhteyden luominen ja soketin kuuntelu	22
6.3.4	Sarjaportista lukeminen	24
6.3.5	TCP-sanoman lähetys	25
7	Yhteenveto.....	26
	Lähteet.....	27
	Liitteet	

Lyhenteet ja käsitteet

ANSI	(American National Standards Institute) Yhdysvaltalainen voittoa tavoittelematon yksityinen organisaatio, joka valvoo erilaisten standardien kehittymistä Yhdysvalloissa
ARM	(Advanced RISC Machines) 32-bittinen mikroprosessoriarkkitehtuuri
ASCII	(American Standard Code for Information Interchange) 128 merkkipaikan laajuinen tietokoneiden merkkistö
DNS	(Domain Name System) Internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteiksi
GCC	(GNU Compiler Collection) GNU-projektin kääntäjien kokoelma
GDB	(GNU Debugger) GNU-projektin virheenjäljitysohjelma
GPIO	(General Purpose I/O) yleiskäyttöinen portti mikro-ohjaimissa ja mikroprosessoreissa
HAL	(Hardware Abstraction Layer) ohjelmiston alajärjestelmä laitteiston abstraktiosta
HART	(Highway Addressable Remote Transducer) protokolla automaation digitaaliseen tiedonsiirtoon
HTTP	(Hypertext Transfer Protocol) protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon
IEEE	(Institute of Electrical and Electronics Engineers) Kansainvälinen tekniikan alan järjestö
IP	(Internet Protocol) TCP/IP-mallin Internet-kerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä Internet-verkossa
ISO	(International Organization for Standardization) kansainvälinen standardisoimisjärjestö

I/O	(Input/Output) tarkoittaa tiedon siirtämistä tai signaloimista tietokonelaitteiston komponenttien välillä
JSON	(JavaScript Object Notation) yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen
OSI-malli	(Open Systems Interconnection Reference Model) kuvaa tiedonsiirtoprotokollien yhdistelmän seitsemässä kerroksessa
PSK	(Phase Shift Keying) eksponentiaalinen modulaatiomenetelmä
RAM	(Random Access Memory) luku-/kirjoitusmuisti on lukemiseen ja kirjoittamiseen käytettävä muistityyppi
RS-232	(Recommended Standard 232) kahden tietokonelaitteen väliseen tietoliikenteeseen tarkoitettu tietoliikenneportti
SPI	(Serial Peripheral Interface bus) synkronoitu sarjaväylä, missä tieto liikkuu bitteinä kahden tai useamman laitteen välillä
SSID	(Service Set Identifier) langattoman lähiverkon verkkotunnus
TCP	(Transmission Control Protocol) tietoliikenneprotokolla, jolla luodaan yhteyksiä tietokoneiden välille, joilla on pääsy Internetiin
TCP/IP	(Transmission Control Protocol / Internet Protocol) usean Internet-liikennöinnissä käytettävän tietoverkko-protokollan yhdistelmä
UART	(Universal Asynchronous Receiver Transmitter) sarjaliikennepiiri on tyypillisesti mikropiiri, joka muuntaa rinnakkaismuotoista tietoa sarjamuotoiseksi ja päinvastoin.
USART	(Universal Synchronous/Asynchronous Receiver-Transmitter) sarjaliikenteen lähetys- ja vastaanotto- piiri, joka muuntaa rinnakkaisdatan sarjadataksi

WEP	(Wired Equivalent Privacy) ensimmäinen työaseman ja tukiaseman välistä langatonta tietoliikennettä suojaamaan kehitetty salausmenetelmä
WLAN	(Wireless Local Area Network) Langaton lähiverkkotekniikka, jolla erilaiset verkkolaitteet voidaan yhdistää ilman kaapeleita
WPA	(Wi-Fi Protected Access) WLAN-verkoissa käytettävä salausprotokolla
WPA2	(Wi-Fi Protected Access II) langattomien 802.11 -verkkojen viimeisin tietoturvastandardi

1 Johdanto

Tämän opinnäytetyön tarkoituksena on kehittää ohjelmisto langattoman tiedonsiirron mahdollistavalle liitäntäyksikölle.

Liitäntäyksikön tehtävänä on lukea mittaus- ja tilatietoja sarjaportin kautta ja lähettää ne reitittimelle WLAN-verkon yli. Tämän jälkeen reititin tallettaa vastaanotetut tiedot pilvipalvelimelle. Opinnäytetyö tehtiin Devecto Oy:lle asiakasprojektina.

Liitäntäyksikkö, johon opinnäytetyössä ohjelmistoa kehitettiin, käyttää STMicroelectronicsin 32-bittistä mikrokontrolleria ja WLAN-moduulia. Ohjelmistokehitysprosessin apuna käytettiin STMicroelectronicsin STM32CubeMX-ohjelmistoa, jolla pystytään generoimaan mikrokontrollerin alustukset graafista käyttöliittymää hyödyntäen.

Opinnäytetyössä tehty ohjelmisto kehitettiin System Workbench for STM32-kehitysympäristössä. System Workbench on ohjelmointiympäristö, johon on integroitu GCC-kääntäjä, GDB-debuggeri, Eclipsen ohjelmointiympäristö ja ST-LINK-tuki. System Workbench valittiin tämän projektin ohjelmointiympäristöksi, koska siinä on kattava tuki STMicroelectronicsin mikrokontrollerille, se on ilmainen eikä siinä ole rajoituksia koodin koolle.

Tässä opinnäytetyössä ohjelmisto toteutettiin aluksi Nucleo-evaluointikortilla, johon oli liitetty WLAN-laajennuskortti. Molemmat kortit ovat STMicroelectronicsin kehittämiä ja valmistamia. Evaluointikorteille tehty ohjelmisto muokattiin ja siirrettiin asiakkaan varsinaiselle liitäntäyksikkökortille sen valmistuttua.

2 WLAN

WLAN tai wireless local area network on langattomaan tiedonsiirtoon tarkoitettu lähiverkkotekniikka, joka mahdollistaa nopean tiedonsiirron verkkoon liitettyjen laitteiden välille. Tämän ansiosta WLAN on suosittu langaton tekniikka kotitalouksissa, toimistoissa ja teollisuudessa. [1]

Puhekielessä WLAN:sta yleensä käytetään ”Wi-Fi”-nimeä, vaikka se oikeasti on Wi-Fi Alliancen tavaramerkki. Tavaramerkin omaavat laitteet ovat läpäisseet yhteensopivuustestit ja ovat saaneet Wi-Fi Alliancelta sertifiointin siitä. Sertifiointi kattaa 2,5 GHz ja 5,0 GHz taajuudella toimivat laitteet ja ne tukevat IEEE 802.11-standardia. [2]

2.1 IEEE-standardi

IEEE 802.11:llä tarkoitetaan standardeja, jotka määrittelevät WLAN:in toiminnan. Alkuperäisen standardin julkaisi Institute of Electrical and Electronics Engineers vuonna 1997, ja se kattoi 2 Mbps ja 1 Mbps tiedonsiirtonopeudet. Standardiin on tehty lukuisia laajennuksia sen julkaisemisen jälkeen ja jokaista laajennusta varten on julkaistu uusi 802.11-protokolla. Näistä protokollista yleisimpiä ovat IEEE 802.11a, joka määrittelee 5,0 GHz kaistanleveyden ja IEEE 802.11b, joka määrittelee nopeammat siirtonopeudet alkuperäiseen protokollaan nähden. Taulukkoon 1 on kirjattu yleisimmät IEEE 802.11-standardit sekä niiden nopeudet, kantamat, taajuusalueet ja siirtonopeudet. Nykyinen standardi on ANSI:n hyväksymä ja ISO:n adoptoima. [3]

IEEE 802.11 Family

Protocol	Release Data	Freq.	Rate (typical)	Rate (max)	Range (indoor)
Legacy	1997	2.4 GHz	1 Mbps	2Mbps	?
802.11a	1999	5 GHz	25 Mbps	54 Mbps	~30 m
802.11b	1999	2.4 GHz	6.5 Mbps	11 Mbps	~30 m
802.11g	2003	2.4 GHz	25 Mbps	54 Mbps	~30 m
802.11n	2008	2.4/5 GHz	200 Mbps	540 Mbps	~50 m

21

Taulukko 1. IEEE 802.11-standardiperhe [4]

Standardin julkaisu ei ollut kumminkaan täysi onnistuminen, sillä sen tietoturvassa oli haavoittuvuus. Vaikka standardissa oli kokonainen kappale omistettu tietoturvalle ja yksityisyydelle, on silti yleistä tietoa, että standardin haavoittuvuus johtui sen suunnittelusta. Standardin julkaisun jälkeen tietoturvaa on paranneltu ja sen pahimpia haavoittuvuuksia on korjattu. [3]

2.2 Kerrosmalli

Kerrosmalli on vuonna 1983 hyväksytty standardi, joka määrittelee tietoliikenteen arkkitehtuurin siten, että se jaetaan seitsemään eri kerrokseen, joista jokainen kerros pystyy kommunikoimaan yhtä ylemmän tai alemman kerroksen kanssa. Kerrosmallin etuna on se, että siitä voidaan vaihtaa yksittäinen kerros ilman, että se häiritsisi muitten kerrosten toimintaa. Kerrosmallissa kerrokset menevät seuraavanlaisesti: [5]

1. fyysinen kerros
2. siirtoyhteyserros
3. verkkokerros
4. kuljetuserros
5. istunterros

6. esitystapakerros

7. sovelluskerros

Kuten kaikki IEEE 802-standardit, myös 802.11-standardi keskittyy kerrosmallin kahdelle alimmalle kerrokselle, fyysiselle kerrokselle ja siirtokerrokselle. [6]

2.2.1 Fyysinen kerros

Fyysinen kerros on OSI-mallin ensimmäinen kerros, ja sen tehtävänä on toimia välittäjänä, jonka kautta lähetin-vastaanottimella lähetetään ja vastaanotetaan viestejä. Kerroksella muutetaan lähetettävä data biteiksi moduloimalla ja vastaanotetut bitit muunnetaan takaisin käytettäväksi dataksi demoduloimalla. Modulointi ja demodulointitapa valitaan käytetyn standardin myötä. [7]

Nykyään IEEE 802.11-standardi käyttää useita eri modulointitapoja kuten:

- Complementary coded keying (CCK)
- Quadrature phase shift keying (QPSK)
- Orthogonal frequency division multiplexing (OFDM)
- Binary phase shift keying (BPSK)
- Quadrature amplitude modulation (QAM).

IEEE 802.11b käyttää CCK- ja QPSK -modulointia ja IEEE 802.11a/g käyttää OFDM-, BPSK-, QPSK- ja QAM-modulaatiota. Alkuperäisessä IEEE 802.11-standardissa on myös tuki frequency hopping spread spectrum (FHSS) -modulaatiolle, mutta sen käyttö on nykyään jäänyt vähemmälle. [8]

2.2.2 Siirtokerros

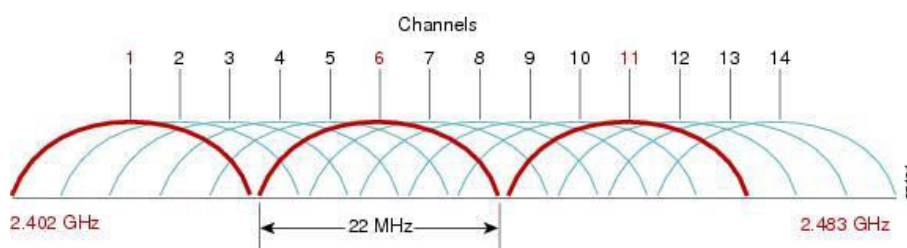
Siirtokerros on OSI-mallin toinen kerros, ja sen tehtäviin kuuluu toimia välittäjänä laitteiston ja ylempien kerrosten välillä. Kerros on vastuussa tietovirran muuttamisesta bitti kerrallaan signaaleiksi ja niiden lähettämisestä laitteiston kautta. Vastaanottavassa

päässä siirtokerros taas kerää vastaanotetut signaalit ymmärrettävään muotoon ja siirtää ne ylemmän kerroksen käsiteltäväksi. [9]

Siirtokerros koostuu kahdesta alakerroksesta: Logical link controlista (LLC) ja Media access controlista (MAC). LLC-alakerros hoitaa ylemmältä kerrokselta saapuvien IP-pakettien kapsuloinnin osoite- ja ohjaustiedoilla. Kapsuloinnin jälkeen paketti, jota kutsutaan kehioksi, siirretään MAC-alakerrokselle, joka muokkaa kehiön fyysiselle kerrokselle sopivaksi ennen sen siirtämistä alemmalle kerrokselle. [7]

2.3 WLAN verkon kanavointi

WLAN-verkot, jotka käyttävät IEEE 802.11-standardia, toimivat useilla eri taajuusalueilla, jotka on jaettu useammalle eri kanavalle. Tällä hetkellä standardi tukee useita eri taajuusalueita, mutta suosituimmat taajuudet ovat 2,4 GHz ja 5,0 GHz. Kuvaan 1 on piirretty kuinka verkon kanavointi tapahtuu 2,4 GHz taajuusalueella. Yksittäisen kanavan kaistanleveys taajuusalueella on noin 20 MHz. Taajuusalueen kanavoinnilla pyritään vähentämään yksittäiselle kaistalle tulevaa kuormaa. [10]



Kuva 1. Verkon kanavointi 2.4 GHz taajuusalueella [11]

3 HART-protokolla

HART-protokolla on teollisuuden mittaus- ja ohjaussovelluksia varten suunniteltu kommunikointiprotokolla. Protokolla kommunikoi käyttämällä analogisia 4-20 mA virtaviestejä, sekä niiden päälle moduloituja digitaalisia signaaleja. Nämä virtaviestit eivät ole itsessään lukukelpoisia, joten vastaanottavan pään täytyy pystyä suodattamaan niistä digitaalinen signaali pois ennen viestin lukemista. HART-protokollaa kutsutaan hybridi protokollaksi koska se perustuu analogiseen ja digitaaliseen viestintäteknikkaan. [12]

Protokollan yksi suurimmista eduista löytyy sen käyttämästä digitaalisesta signaalista. Normaalisti pelkällä analogisella virtaviesteillä voitaisiin kommunikoida vain yhteen suuntaan, mutta digitaalinen signaali mahdollistaa kaksisuuntaisen viestinnän. Digitaalinen signaali myös mahdollistaa sen että laitteelta voidaan saada muutakin dataa kuin mittaustietoa. [12]

3.1 Viestien rakenne

Protokollan pyyntöviestin pakettien rakenne pysyy aina samana ja se sisältää pakollisia ja valinnaisia tietokenttiä. Paketit sisältävät seuraavat viestikentät: [13]

1. johdanto, auttaa vastaanottajaa synkronisoitumaan merkkijonoon
2. aloitusbitti, ilmoittaa viestin tyyppin
3. osoite, sisältää isäntälaitteen ja orjalaitteen osoitteet.
4. laajennus, määrittelemätön kenttä
5. komento, sisältää laitteelle menevän HART komennon
6. databittien määrä, ilmoittaa kuinka monta bittiä on tila- ja datakentissä yhteensä
7. tila, orjalaitteelle tarkoitettu kenttä, joka sisältää laitteen tilan sekä mahdolliset kommunikointi virheet
8. data, sisältää orjalaitteelle menevän datan

9. tarkiste, käytetään kommunikointivirheiden etsintään

3.2 Laitteiden yhteensopivuus

Laitteiden yhteensopivuudella tarkoitetaan sitä, että HART-protokollaa tukevat laitteet pystyvät kommunikoimaan keskenään niiden valmistajasta huolimatta. Protokolla sisältää noin 40 eri common practise- ja universal-komentoa, jotka tulisi löytyä jokaisesta HART-yhteensopivasta laitteesta. [12]

4 STMicroelectronics

STMicroelectronics on yksi maailman suurimpia puolijohhteita valmistava yritys. Yritys on muodostunut vuonna 1987 SGS Microelettronican ja Thomson Semiconducteurs:in fuusiosta. STMicroelectronics on johtava teknologinen innovaattori, jolla on noin 17 000 patenttia ja 9 400 henkilöä työskentelemässä tutkimuksen ja tuotekehityksen parissa. Pelkästään vuonna 2017 yritys jätti 500 uutta patenttihakemusta. [14]

STMicroelectronics tarjoaa myös valmiita teknologisia ratkaisuja älykkäisiin autoihin, joilla ajamisesta saadaan turvallisempaa ja ekologisempaa. Näihin ratkaisuihin kuuluu advanced driver assistance systemsin tuoteperhe, joka sisältää kameroita, tutkia sekä muita ajoneuvon ympäristöä monitoroivia laitteita. [14]

Yrityksen valikoimasta löytyy myös kehitysalustoja, joilla voi evaluoida mikroprosessorien ja WLAN-moduulien toimivuutta. Näitä kehitysalustoja käytettiin tässä työssä. [15]

4.1 Wi-Fi -moduulit

STMicroelectronicsin Wi-Fi -moduulit on rakennettu IEEE 802.11-standardia tukevan lähetin-vastaanottimen ympärille. Moduuleihin on myös integroituna tehovahvistimet, 32-bittiset mikrokontrollerit, kellogeneraattorit ja jännitteensäätimet. Moduuleita on saatavilla integroidulla antennilla sekä U.FL-liittimellä. [15]

Moduulit ovat pienikokoisia ja vähävirtaisia, joten ne sopivat kiinteisiin ja liikuteltaviin laitteisiin. Ne myös helpottavat laitteiden kehitystä, sillä niissä on sisäänrakennettuna TCP/IP-pino. Moduulit ovat ideaalisia, kun tuotekehityksessä nopeus on tärkeää. [15]

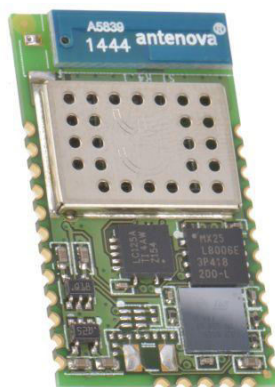
Alla olevissa kappaleissa on käsitelty kahden eri moduulin merkittävimpiä eroja.

4.1.1 SPWF01SA-moduuli

SPWF01SA-moduulin (kuva 2) tarkemmat tiedot: [16]

- STM32 ARM Cortex-M3 mikrokontrolleri

- 64 KB RAM ja 512 KB flash-muistia
- saatavilla myös 1.5 MB flash-muistilla
- tukee WEP, WPA ja WPA2 autentikointia
- 16 konfiguroitavaa GPIO-pinniä
- koko: 26,29 x 15,24 x 2,35 mm
- mahdollisuus päivittää firmware langattomasti sekä UART-rajapinnan kautta
- tiedostojärjestelmän päivitys UART-rajapinnan kautta



Kuva 2. SPWF01SA-moduuli [17]

4.1.2 SPWF04SA-moduuli

SPWF01SA-moduulin (kuva 3) tarkemmat tiedot: [18]

- STM32F439 ARM Cortex-M4-mikrokontrolleri
 - 246 KB SRAM ja 2 MB flash-muistia
- tukee WEP, WPA2 PSK, WPA2 Enterprise-autentikointia
- 21 konfiguroitavaa GPIO-pinniä
- koko: 26,29 x 15,24 x 2,35 mm

- mahdollisuus päivittää firmware ja tiedostojärjestelmä langattomasti sekä UART-rajapinnan kautta



Kuva 3. SPWF04SA-moduuli [19]

4.2 Mikrokontrollerit

STMicroelectronics tarjoaa laajan valikoiman mikrokontrollereita eri tarkoituksiin. Tuotevalikoimasta löytyy erittäin vähävirtaisiin sovelluksiin 8-bittisistä kontrollereita ja paljon laskentatehoa tarvitseviin sovelluksiin 32-bittisiä kontrollereita. [20]

4.3 Kehitysalustat

Kehitysalustojen tehtävänä on antaa mahdollisuus evaluoida mikrokontrollereiden ja muiden oheislaitteiden toimintaa ennen varsinaisen tuote- tai ohjelmistokehityksen aloittamista. Alustoihin löytyy myös paljon STMicroelectronicsin valmistamia esimerkki-sovelluksia, joita voi hyödyntää evaluoinnissa. [21]

Alla olevissa kappaleissa käsitellään vain tässä työssä käytettyjä kehitysalustoja, NUCLEO-L476RG:tä ja X-NUCLEO-IDW01M1:tä.

4.3.1 NUCLEO-L476RG

Tämä kehitysalusta tarjoaa mahdollisuuden evaluoida STM32L476RG-mikrokontrolleria, joka on STMicroelectronicsin 32-bittinen prosessori. Kehitysalustassa on tuki Arduino

Uno V3- ja ST morpho-liitännöille, jotka mahdollistavat lukuisten laajennuskorttien liittämisen kehitysalustalle. Kehitysalustaan on integroitu ST-LINK/V2-1, joka hoitaa alustan ohjelmoinnin ja debuggauksen. Alustalta löytyy myös kaksi painonappia, joista toinen on ohjelmoitavissa ja toinen hoitaa nollauksen, sekä yksi kappale LED-valoja. Kehitysalusta näkyy kuvassa 4. [22]



Kuva 4. NUCLEO-L476RG -kehitysalusta

4.3.2 X-NUCLEO-IDW01M1

Tälle kehitysalustalle on integroitu SPWF01SA WLAN-moduuli ja se toimii laajennuskorttina STMicroelectronicsin nucleo-kehitysalustoille. Alustalle integroitua WLAN-moduulia voidaan ohjata nucleon mikrokontrollerin UART-rajapinnan kautta. Kehitysalusta on yhteensopiva ST morpho- ja Arduino-liitännöiden kanssa. Kehitysalustasta näkyy kuvassa 5. [23]



Kuva 5. X-NUCLEO-IDW01M1 kehitysalusta

5 Ohjelmiston vaatimusmäärittely

Ohjelmiston tulisi vähintään kattaa seuraavat toiminnot:

- yhdistää liitäntäyksikkö WLAN verkon yli reitittimeen
- lukea sarjaportin kautta kiinnitetyltä laitteelta mittaus- ja tilatiedot
- tuki HART-protokollaan
- muuttaa ASCII muotoinen data JSON muotoiseksi
- lähettää luettu data reitittimelle JSON formaatissa

Liitäntäyksikön ja reitittimen välinen kommunikointi toteutetaan TCP socket-yhteyden avulla. Liitäntäyksikön ohjelmistoon tulee toteuttaa TCP server toiminnot sanomien vastaanottamiseen sekä TCP client sanomien lähettämiseen. Liitäntäyksikön liikkointiparametrien konfigurointi tulee toteuttaa joko yksikölle toteutettavan HTTP palvelimen kautta tai tietokoneella sarjaportin kautta.

5.1 JSON-formaatti

JSON on kevyt, datan siirtoon tarkoitettu tekstiformaatti. JSON-tekstiformaatti käyttää JavaScriptin syntaksia, mutta se on silti ohjelmointikielestä riippumaton, joka tarkoittaa sitä, että tekstiä voidaan lukea ja kirjoittaa millä ohjelmointikielellä hyvänsä.

5.2 Sarjaportin toiminnan määrittelyn tarkennus

Liitäntäyksikkö pyytää RS-232 sarjaporttiin kytketyltä laitteelta tila- tai mittaustietoja lähettämällä ASCII muotoisen komennon sarjaportin kautta. Vastauksena saadut JSON-muotoiset tiedot lähetetään eteenpäin WLAN-yhteyden kautta reitittimelle.

5.3 HART-tuen määrittelyn tarkennus

Liitäntäyksikkö lähettää HART-protokollan mukaisia sanomia väylään kytketylle laitteelle. Vastauksena saadut viestit muutetaan ASCII muodosta JSON muotoon ennen kuin ne lähetetään eteenpäin WLAN-yhteyden kautta pilvipalvelimelle.

5.4 WLAN-määrittelyn tarkennus

Liitäntäyksikön ohjelmiston TCP-serveri kuuntelee TCP-porttia, johon pilvipalvelimesta lähetetään tiedonlukupyyntöjä. Liitäntäyksikkö hakee sarjaväylään kiinnitetyltä laitteelta tiedot ja lähettää ne TCP clientin kautta reitittimelle.

WLAN:in konfigurointi voidaan suorittaa HTTP-palvelimella pyörivien sivustojen kautta tai tietokoneella sarjaportin kautta. Konfiguroitavia parametreja ovat:

- reitittimen SSID
- autentikointimenetelmä
- autentikoinnin salasana
- IP-osoite
- aliverkon peite
- yhdyskäytävä
- DNS osoite.

Konfiguraatio tallennetaan WLAN-moduulin flash-muistiin.

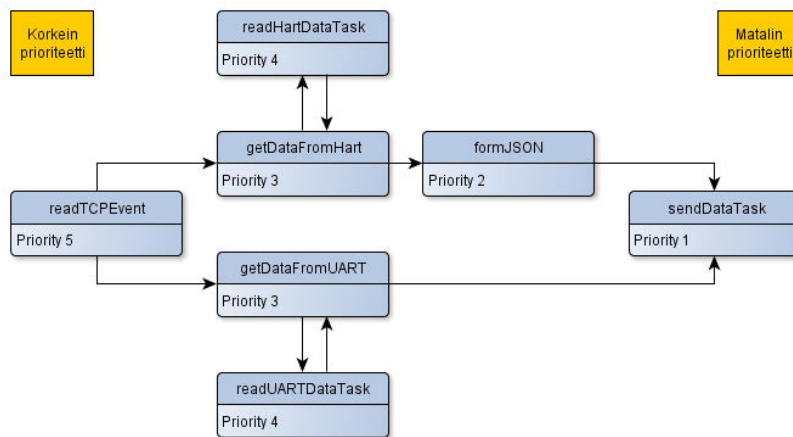
Liitäntäkortin WLAN voidaan asettaa konfigurointitilaan erillisen kytkimen tai painonapin avulla.

5.5 Käyttöjärjestelmä

Liitäntäyksikön ohjelmisto rakennetaan FreeRTOS-käyttöjärjestelmän ympärille. Käyttöjärjestelmä tekee ohjelmistosta tapahtumavetoisen ja sitä kautta tehokkaamman.

Käyttöjärjestelmän toiminta perustuu tehtäviin, joita voidaan käynnistää tietyin aikavälein tai ulkoisten keskeytysten tapahtuessa. Tehtävät voidaan jakaa keskenään eri prioriteettien mukaan, jotka ohjaavat sitä, mikä tehtävä lähtee milloinkin käyntiin. Tehtävien välinen kommunikointi pyritään hoitamaan pääsääntöisesti ilmoitusten avulla RAM muistin säästämiseksi.

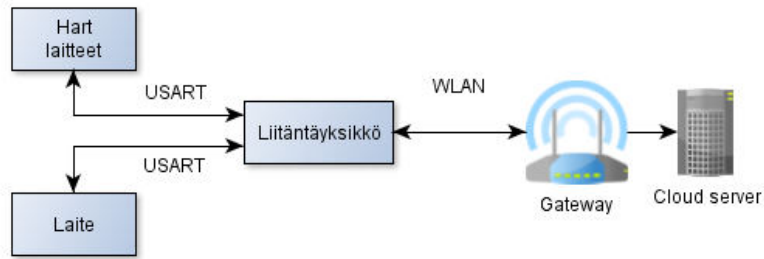
Yksinkertaistettu kuvaus liitäntäyksikön tehtävistä ja niiden prioriteeteista on piirretty kuvaan 6.



Kuva 6. Käyttöjärjestelmän tehtävät

5.6 Ohjelmiston arkkitehtuuri

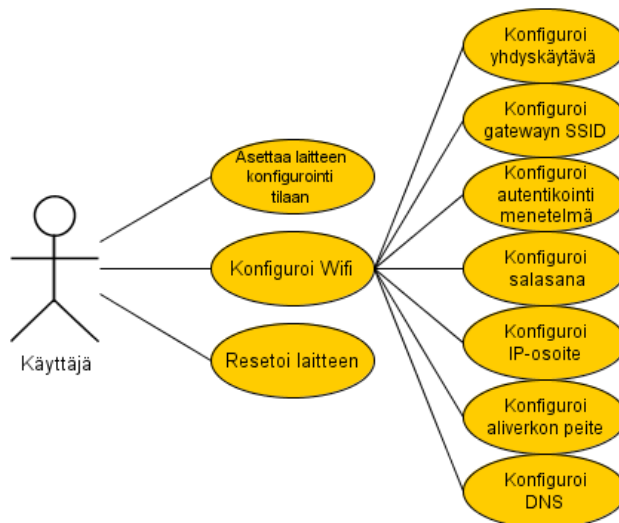
Ohjelmisto on osa järjestelmää, jonka yksinkertaistettu arkkitehtuuri on nähtävissä kuvasta 7. Arkkitehtuurissa otetaan huomioon vain ohjelmistoon liittyvät komponentit ja sen tarkoitus on kuvata liitäntäyksikön roolia järjestelmässä.



Kuva 7. Arkkitehtuuri

5.7 Käyttötapaukset

Kappaleessa esitellyt käyttötapaukset koskevat vain ja ainoastaan liitäntäyksikköä. Käyttäjällä ei ole juuri muita tehtäviä wifin konfiguroinnin lisäksi, koska liitäntäyksikkö hoitaa automaattisesti tiedonkeruun ja lähetyksen. Käyttötapauskaavio kuvaa käyttäjän roolia liitäntäyksikön toiminnassa. Käyttötapauskaavio on piirretty kuvaan 8.



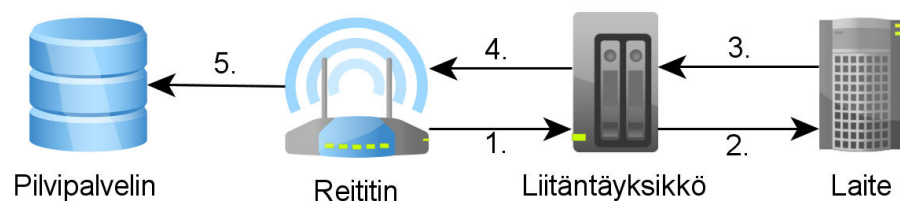
Kuva 8. Käyttötapauskaavio

5.8 Ohjelmiston logiikka

Ohjelmiston päätehtävänä on vastata pilvipalvelimelta tuleviin tiedonlukupyyntöihin hakemalla sarjaporttiin kytketyltä laitteelta mittaus- ja tilatiedot ja lähettää ne takaisin pilvipalvelimelle. Prosessi menee seuraavanlaisesti:

1. Reititin lähettää tiedolukupyynnön liitäntäyksikön TCP serverille.
2. Liitäntäyksikkö pyytää sarjaporttiin kytketyltä laitteelta mittaus- ja tilatiedot ja alkaa odottamaan viestiä sarjaväylältä.
3. Laite vastaa liitäntäyksikön viestiin lähettämällä sarjaportin kautta mittaus- sekä tilatiedot.
4. Liitäntäyksikkö lukee laitteelta saadut tiedot ja lähettää ne TCP clientin kautta reitittimelle.
5. Reititin lähettää saadut tiedot pilvipalvelimelle.

Ohjelmiston toimintalogiikka on esitetty graafisesti kuvassa 9. Ohjelmiston toimintalogiikka kuvaa laitteen toimintaa järjestelmässä.



Kuva 9. Ohjelmiston logiikka

6 Toteutus

Tässä työssä toteutettiin ohjelmisto langattoman tiedonsiirron mahdollistavaan liitäntäyksikköön, joka käytti STMicroelectronicsin SPWF01SA-moduulia. Ohjelmisto aluksi toteutettiin STMicroelectronicsin NUCLEO-L476RG kehitysalustalle, johon oli kytketty X-NUCLEO-IDW01M1 WLAN-laajennuskortti. Kuvassa 10 on kehitysalustat yhdistettynä.



Kuva 10. Kehitysalustat yhdistettynä

Työ aloitettiin tutustumalla kehitysalustan ja WLAN-laajennuskortin toimintoihin datasivujen avulla. Ennen ohjelmiston toteutusta WLAN-moduulin firmware päivitettiin versioon 3.5.3 ja opeteltiin toimintoja lähettämällä sille komentoja tietokoneen sarjaportin kautta. Sarjaportin terminaaliohjelmistona käytin tera termiä. Tutkimusvaiheen ja WLAN-moduulin toiminnan selvittämisen jälkeen aloitettiin ohjelmiston toteutuksen suunnitteluvaihe kehitysalustalle.

Kehitysalustalla ei ollut HART-modeemia, joten sen toiminnallisuutta ei voitu kehitysalustan avulla toteuttaa. Työssä saatiin valmiiksi HART-tukea ja langatonta konfigurointia lukuun ottamatta muut toiminnallisuudet.

6.1 Työkalut

Kehitysalustana työssä oli käytössä STMicroelectronicsin valmistama NUCLEO-L476RG. Kehitysalusta käytti STM32L476RG 32-bittistä mikrokontrolleria. Kontrollerin ohjelmointi ja debuggaus tapahtui ST-LINKillä, joka oli integroituna kehitysalustalle.

Kehitysalusta soveltui hyvin tähän työhön, vaikka siitä puuttuikin yhtä UART-liitäntää lukuunottamatta liitännät eri rajapinnoille.

Kehitysalustan lisäksi käytössäni oli WLAN-laajennuskortti X-NUCLEO-IDW01M1. Laajennuskortin tehtävänä oli yhdistää SPWF01SA WLAN-moduuli kehitysalustaan siten, että sitä voitaisiin käskyttää SPI- ja UART-rajapintojen kautta. Kortin laiteohjelman päivitys tapahtui myös kehitysalustan UART-rajapinnan kautta. Laajennuskortilla oli samat liitäntämahdollisuudet kuin kehitysalustalla, joten niiden väliselle liitännälle ei tarvinnut tehdä yhtään muutoksia tai lisäyksiä.

Itse ohjelman kirjoittamiseen käytettiin system workbench for STM32 -nimistä ohjelmistoa, joka on kolmannen osapuolen tarjoama ilmainen ohjelmointiympäristö STMicroelectronicsin mikrokontrollereille. Ohjelmointiympäristö perustuu vahvasti eclipsen ohjelmointiympäristöön ja se sisältää kaikki tarvittavat työkalut ohjelmien kirjoittamista, kääntämistä ja debuggausta varten.

Mikrokontrollerin alustukset sekä kehitysprojektin perusasetukset luotiin STM32CubeMX ohjelmiston avulla, joka on STMicroelectronicsin tarjoama ohjelmistonkehityksen työkalu. Ohjelmiston avulla voidaan luoda valmiit alustukset ja projektitiedostot STMicroelectronicsin mikrokontrollereille graafista käyttöliittymää hyödyntäen. STM32CubeMX:llä on myös mahdollisuus lisätä väliohjelmistoja sekä HAL-kirjastoja luotavaan projektiin.

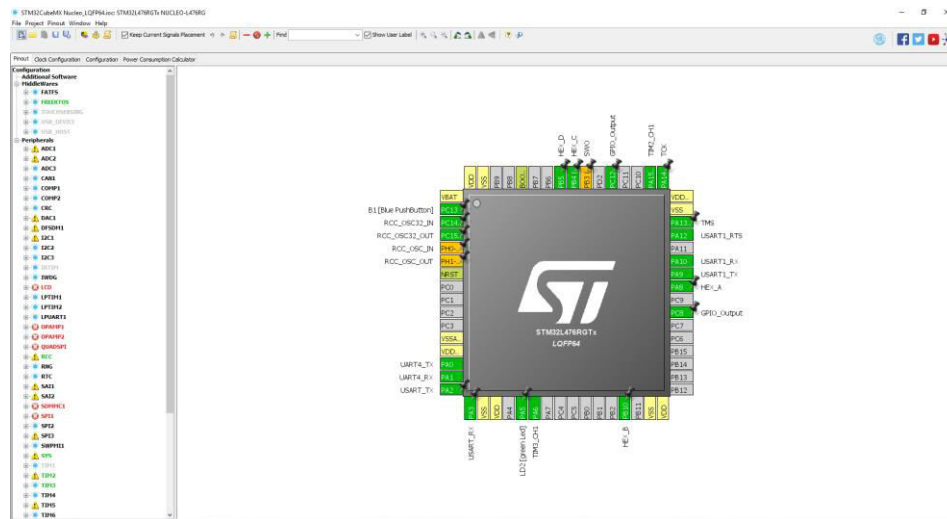
6.2 Käytetyt väliohjelmistot ja HAL-kirjastot

Ohjelmiston toteutuksessa käytettiin STMicroelectronicsin tarjoamaa X-CUBE-WIFI1 väliohjelmistoa, joka toimii ohjelmistorajapintana WLAN-moduulin ja liitäntäyksikön ohjelmiston välillä. Väliohjelmiston sisältämät funktiokutsut hoitavat WLAN-moduulille menevien käskyjen lähetyksen ja vastausten lukemisen. Väliohjelmiston funktiot hyödyntävät mikrokontrollerin HAL-kirjastoa viestimisessä.

Myös STM32L476RG mikrokontrollerille tarkoitetut HAL-kirjastot lisättiin ohjelmistoon, koska väliohjelmisto on riippuvainen niistä ja ne helpottavat kontrollerin ohjelmiston tekemistä. HAL-kirjasto sisältää funktiokutsut, joilla saadaan alustettua mikrokontrollerin I/O-portit, kellot, keskeytysrutiinit, GPIO-pinnit sekä ytimen kellotaajuuden. Viestien lähettäminen ja vastaanottaminen I/O-porttien kautta tapahtuu myös HAL-kirjaston funktioiden avulla.

6.3 Ohjelmistonkehitys ja ohjelmointi

Ohjelmistonkehitysprosessi aloitettiin lataamalla tarvittavat ohjelmistot STMicroelectronicsin sivustolta. STM32CubeMX ohjelmistolla luotiin ohjelmistokehitysympäristön projekti ja alustukset kehitysalustan mikrokontrollerille. Ohjelman käyttöliittymä näkyy kuvassa 11.



Kuva 11. Kuvakaappaus STM32CubeMX ohjelmistosta

Ohjelmiston avulla luotiin alustukset UART4-, USART1- ja USART2-rajapinnoille, TIM2- ja TIM3-kelloille, kehitysalustan painonapeille, ulkoisille kide oskilaattoreille sekä useille GPIO-pinneille. Myös HAL-kirjastot ja FreeRTOS-käyttöjärjestelmä lisättiin projektiin. UART4-rajapinta käytetään WLAN-moduulin tiedonsiirtoon, USART2-rajapintaa käytetään debug-viestien lähettämiseen tietokoneen sarjaporttiin ja USART1-rajapintaa käytetään kytketyn laitteen kanssa viestimiseen. Tarkemmat alustukset on kirjattu taulukkoon 2.

Pinni	Toiminto	Nimike
PC13	GPIO_EXTI13	B1[Blue PushButton]
PC14-OSC32_IN (PC14)	RCC_OSC32_IN	
PC15-OSC32_OUT (PC15)	RCC_OSC32_OUT	
PH0-OSC_IN (PH0)	RCC_OSC_IN	
PH1-OSC_OUT (PH1)	RCC_OSC_OUT	
PA0	UART4_TX	
PA1	UART4_RX	
PA2	USART2_TX	USART_TX
PA3	USART2_RX	USART_RX
PA5	GPIO_Output	LD2 [green Led]
PA6	TIM3_CH1	
PB10	GPIO_Input	HEX_B
PC8	GPIO_Output	
PA8	GPIO_Input	HEX_A
PA9	USART1_TX	
PA10	USART1_RX	
PA12	USART1_RTS	
PA13 (JTMS-SWDIO)	SYS_JTMS-SWDIO	TMS
PA14 (JTCK-SWCLK)	SYS_JTCK-SWCLK	TCK
PA15 (JTDI)	TIM2_CH1	
PC12	GPIO_Output	
PB3(JTDO-TRACESWO)	SYS_JTDO-SWO	SWO
PB4 (NJTRST)	GPIO_Input	HEX_C
PB5	GPIO_Input	HEX_D

Taulukko 2. Mikrokontrollerin alustukset

6.3.1 WLAN-moduulin alustus ja käynnistys

Ohjelmistokehitysympäristön projektin ja alustusten luonnin jälkeen aloitettiin ensimmäisen toiminnallisuuden kirjoittaminen ohjelmistoon, joka oli moduulin alustaminen ja käynnistäminen. Tämä tapahtui kutsumalla väliohjelmiston `wifi_init` funktiokutsua ja antamalla kuvan 12 mukaiset parametrit.

```

/* wifi init parameters */
wifi_config config;
config.power=wifi_active;
config.power_level=high;
config.dhcp=on; //use DHCP IP address
config.mcu_baud_rate = 115200;

/* Init the wi-fi module */
status = wifi_init(&config);

```

Kuva 12. Moduulin alustus ja käynnistys

Ohjelmistokoodia voitiin testata ajamalla ohjelmisto kehitysalustalle sisään ja debuggaamalla sitä. Debugatessa huomattiin, että ohjelmiston ajo keskeytyi wifi_init funktion sisällä, koska kehitysalusta ei saanut luettua WLAN-moduulin vastauksia.

Ongelmaan etsittiin ratkaisua vertaamalla tehtyjä alustuksia esimerkki-ohjelmiston alustuksiin. Vertailemalla huomattiin että kellojen ja keskeytysrutiinien alustukset erosivat toisistaan, joten alustukset muutettiin vastaamaan esimerkin alustuksia. Tämä ei vielä korjannut tilannetta, joten ongelman syyn etsintää jatkettiin tutkimalla projektin asetuksia. Tutkimuksessa huomattiin, että projektin esikäntäjän asetukset erosivat esimerkin asetuksista, joten myös esikäntäjän asetuksia muutettiin vastaamaan esimerkkiprojektia. Tämäkään ei kuitenkaan korjannut asiaa, joten ongelman selvittelyä jatkettiin, jotta löydettäisiin ratkaisu, joka mahdollistaisi työn jatkamisen.

Lukuisten eri konfiguraatioiden ja testien jälkeen päädyttiin johtopäätökseen, että WLAN-modulin HAL-kirjasto ei ollut kaikilta osin yhteensopiva STM32CubeMX:llä luodun projektin kanssa. Ongelma näytti ainakin osittain liittyvän FreeRTOS-käyttöjärjestelmän ja WLAN-modulin käyttämän kellotimerin eroavaisuuksiin.

Koska työhön käytettävissä oleva aika oli rajallinen ja perustoiminnot pitäisi saada toteutettua, päätettiin kokeilla vaihtoehtoista toteutustapaa. STM32CubeMX:llä luotu projekti hylättiin kokonaan ja uudeksi toteutustavaksi valittiin ohjelmiston rakentaminen WLAN-modulin esimerkkiprojektin perustalle, vaikka tämä tarkoittikin sitä, että FreeRTOS-käyttöjärjestelmästä piti luopua.

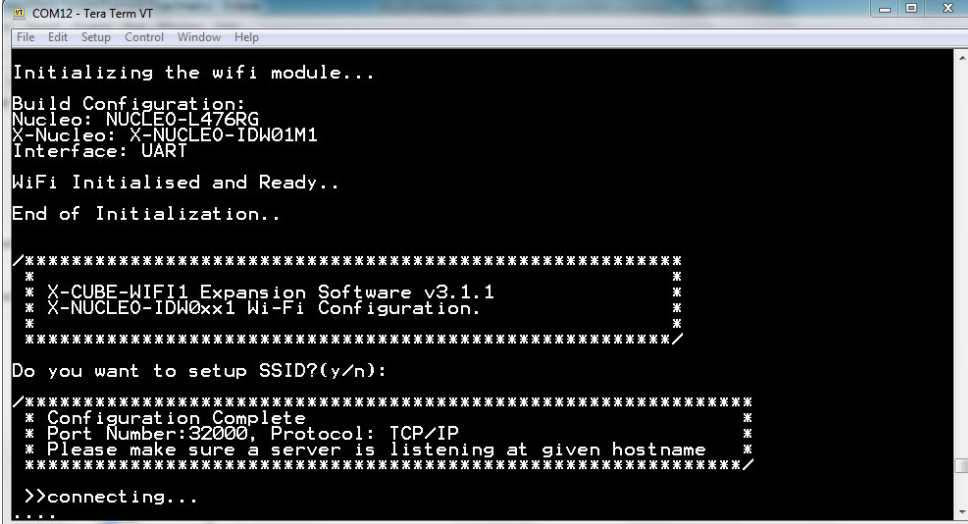
Esimerkkiprojektin työstäminen aloitettiin poistamalla tarpeeton koodi ohjelmasta ja varmistamalla, että WLAN-moduulin saa alustettua ja käynnistettyä. Tällä toteutustavalla WLAN-moduuli saatiin viimein käynnistymään ja voitiin jatkaa ohjelman muiden toimintojen toteuttamista.

6.3.2 WLAN-moduulin konfigurointi sekä verkkoon liittäminen

WLAN-moduulin konfiguroinnin toteutustavaksi valittiin konfigurointiparametrien lähetyksen sarjaportin kautta. Kehitysalustan käynnistyksen yhteydessä sarjaporttiin lähetetään käyttäjälle kysymys haluaako hän konfiguroida laitteen. Jos käyttäjä vastaa ”y”, niin alusta alkaa kysellä käyttäjältä konfigurointi parametrejä, kuten SSID:tä, autentikointitapaa, salasanaa ja reitittimen IP-osoitetta. Mikäli käyttäjä vastaa ”n” tai ei vastaa tietyn ajan kuluessa, niin kehitysalusta käyttää oletusparametrejä tai aikaisemmassa kyselyssä lähetettyjä parametrejä.

Parametrien kyselyn ja WLAN-moduulin konfiguroinnin jälkeen ohjelmisto skannaa ensimmäiseksi alueella olevat WLAN-verkot ja yrittää sitten liittyä konfiguroituun verkkoon jos se löytyy, muuten skannataan uudelleen. Verkkoon liittyminen tapahtuu wifi_connect funktiokutsun avulla, joka tarvitsee parametreikseen vain verkon SSID:n, salasanan ja autentikointitavan. Kehitysalusta yrittää liittyä verkkoon niin kauan kunnes se onnistuu siinä.

Kuvassa 13 näkyy konfiguroinnin ja verkkoon liittämisen aikana tietokoneen sarjaporttiin tulostetut kehitysalustan tilatiedot.



```
COM12 - Tera Term VT
File Edit Setup Control Window Help

Initializing the wifi module...
Build Configuration:
Nucleo: NUCLEO-L476RG
X-Nucleo: X-NUCLEO-IDW01M1
Interface: UART

WiFi Initialised and Ready..
End of Initialization..

/*****
 * X-CUBE-WIFI1 Expansion Software v3.1.1
 * X-NUCLEO-IDW0xx1 Wi-Fi Configuration.
 *****/

Do you want to setup SSID?(y/n):

/*****
 * Configuration Complete
 * Port Number:32000, Protocol: TCP/IP
 * Please make sure a server is listening at given hostname
 *****/

>>connecting...
....
```

Kuva 13. Moduulin konfigurointi ja verkkoon liittäminen

6.3.3 TCP-yhteyden luominen ja soketin kuuntelu

Yhteyden luominen tapahtuu `wifi_socket_client_open` funktiokutsulla, joka ottaa parametreikseen reitittimen IP-osoitteen, portin numeron, TCP-protokollan ja avattavan soketin. Kehitysalusta yrittää avata yhteyttä niin kauan kunnes se onnistuu siinä tai WLAN-yhteys katkeaa. Koodi katkelma yhteyden luomisesta näkyy kuvassa 14.

```
printf("\r\n >>Connecting to socket\r\n");

if(socket_open == 0)
{
    /* Connect to Socket */
    Wifi_Status_t status = WiFi_MODULE_SUCCESS;
    status = wifi_socket_client_open((uint8_t *)console_host, portnumber, (uint8_t *)protocol, &socket_id);

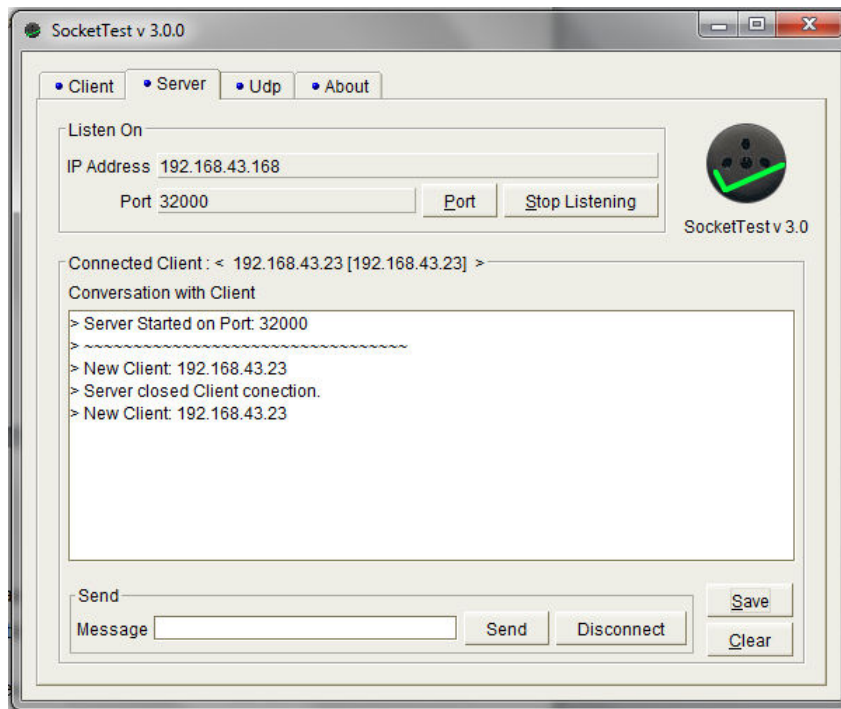
    if(status == WiFi_MODULE_SUCCESS)
    {
        printf("\r\n >>Socket Open OK\r\n");
        printf("\r\n >>Socket ID: %d\r\n", socket_id);
        socket_open = 1;
        wifi_state = wifi_state_socket_listen;
    }
    else
    {
        printf("Socket connection Error");
    }
}
else
{
    printf("Socket already opened!");
}

break;
```

Kuva 14. Yhteyden luonti

Yhteyden luotuaan kehitysalusta alkaa kuuntelemaan avattua sokettia. Sokettiin tulevia TCP-sanomia seurataan keskeytysrutiinien avulla. Kun sokettiin saapuu viesti, kehitysalusta siirtyy lukemaan sarjaporttiin kytketyltä laitteelta tietoja.

Yhteyden luomista ja soketin kuuntelua pystyttiin testaamaan pystyttämällä tietokoneelle soketti palvelin SocketTestV3 ohjelmalla. Kuvassa 15 näkyy kuvakaappaus soketti palvelimesta, johon kehitysalusta on ottanut yhteyden.



Kuva 15. SocketTestV3 ohjelma

6.3.4 Sarjaportista lukeminen

Kehitysalusta lähettää ensimmäiseksi sarjaportin kautta viestin siihen kytketylle laitteelle. Tämän tarkoituksena on tarkistaa, että sarjaporttiin on kytketty laite ja että se on hengissä. Viestin lähettämisen jälkeen kehitysalusta alkaa odottamaan laitteen vastausta. Jos laite ei vastaa viiden sekunnin sisällä tai viestin sisältö ei ole oikeassa muodossa, kehitysalusta yrittää viestin lähettämistä uudelleen. Viestien lähettäminen tapahtuu `HAL_UART_Transmit_IT` funktiokutsulla ja vastaanottaminen `HAL_UART_Receive` funktiokutsulla.

Kun kytketyltä laitteelta tullut viesti ollaan vastaanotettu onnistuneesti, kehitysalusta lähettää seuraavan viestin, jonka tarkoituksena on pyytää kytketyltä laitteelta mittaus- ja tilatietoja. Kun nämä tiedot ollaan vastaanotettu onnistuneesti, kehitysalusta siirtyy lähettämään saatuja tietoja TCP-yhteyden yli reitittimelle. Koodikatkelma mittaus- ja tilatietojen hausta näkyy kuvassa 16.

```

//Status = H, Info = I, Both = M
HAL_UART_Transmit_IT(&uart4, (uint8_t*)"m", 1);
while (uartReady != SET)
{
}
HAL_UART_Receive(&uart4, (uint8_t*)tcuReadBuf, MAX_BUF_SIZE, 5000);
printf("Rx: %s\r\n", (char*)tcuReadBuf);

```

Kuva 16. Mittaus- ja tilatietojen haku

6.3.5 TCP-sanoman lähetys

Sanoma lähetetään `wifi_socket_client_write` funktiokutsun avulla, joka ottaa parametreikseen avatun soketin, lähetettävien merkkien määrän ja taulukon, jossa on lähetettävä data. Kehitysalusta yrittää lähettää sanomaa niin kauan kunnes se onnistuu siinä. Sanoman onnistuneen lähetyksen jälkeen kehitysalusta siirtyy taas kuuntelemaan sokettia. Koodi katkelma sanoman lähetyksestä näkyy kuvassa 17.

```

//If data transmission is completed correctly, move on, else clear buffer and try again
if (strstr((const char*)tcuReadBuf, (const char*)"-End") != NULL)
{
    wifi_state = wifi_state_socket_reply;
    tcu_state = tcu_state_off;
}
else
    memset(tcuReadBuf, 0x00, sizeof(tcuReadBuf));

```

Kuva 17. Sanoman lähetys

7 Yhteenveto

Opinnäytetyön aikana saatiin ohjelmiston perustoiminnallisuudet valmiiksi HART tukea ja langatonta konfigurointia lukuun ottamatta. Ohjelmiston toiminta on toistaiseksi testattu kokonaisuudessaan vain kehitysalustalla. Liityntäkortit saapuivat opinnäytetyön loppuvaiheessa ja niillä ehdittiin kokeilla ohjelmiston toimintaa hieman, mutta ei kaikilta osin. Kehitysalustalle toteutetun ohjelmiston siirtämistä liityntäkorteille on tarkoitus jatkaa varsinaisen opinnäytetyön jälkeen.

Työn aikana suurimmaksi haasteeksi osoittautui WLAN-moduulin toimimaan saaminen. Työtä hankaloitti myös se, että minulla ei ollut paljoa kokemusta STMicroelectronicsin tuotteista eikä sulautettujen laitteiden ohjelmiston kehityksestä.

Opinnäytetyö oli itsessään hyvin opettavainen ja sain paljon kokemusta sulautetuista laitteista, C-ohjelmoinnista ja vianetsinnästä. STMicroelectronicsin valmiit esimerkit osoittautuivat työn loppu- ja keskivaiheilla erittäin hyödyllisiksi.

Lähteet

1. Wikipedia. Wireless LAN (8.4.2018). [Viitattu 9.4.2018] Saatavissa: https://en.wikipedia.org/wiki/Wireless_LAN
2. Webopedia, Vangie Beal. Wi-Fi (wireless networking). [Viitattu 9.4.2018] Saatavissa: https://www.webopedia.com/TERM/W/Wi_Fi.html
3. Cisco, Edgar Danielyan. IEEE 802.11 - The Internet Protocol Journal - Volume 5, Number 1. [Viitattu 9.4.2018] Saatavissa: <https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-21/ieee.html>
4. NetworxSecurity. IEEE 802.11. [Viitattu 19.4.2018] Saatavissa: <http://www.networxsecurity.org/members-area/glossary/i/ieee-80211.html>
5. Koudata. OSI-malli. [Viitattu 10.4.2018] Saatavissa: <http://www.koudata.fi/node/598>
6. TechTarget, Margaret Rouse. IEEE 802 Wireless Standards: Fast Reference (2006). [Viitattu 10.4.2018] Saatavissa: <https://searchmobilecomputing.techtarget.com/definition/IEEE-802-Wireless-Standards-Fast-Reference>
7. Control engineering, Daniel E. Capano. Wi-Fi and the OSI model (18.9.2014). [Viitattu 10.4.2018] Saatavissa: <https://www.controleng.com/single-article/wi-fi-and-the-osi-model/8b71b0494b6b7fd5291856d02e104eb4.html>
8. Tutorialspoint. Wi-Fi - Radio Modulation. [Viitattu 10.4.2018] Saatavissa: https://www.tutorialspoint.com/wi-fi/wifi_radio_modulation.htm
9. Tutorialspoint. DCN – Data-link layer introduction. [Viitattu 10.4.2018] Saatavissa: https://www.tutorialspoint.com/data_communication_computer_network/data_link_layer_introduction.htm
10. Electronicsnotes. Wi-Fi channels, frequencies, bands & bandwidths. [Viitattu 10.4.2018] Saatavissa: <https://www.electronics->

notes.com/articles/connectivity/wifi-ieee-802-11/channels-frequencies-bands-bandwidth.php

11. Meraki. Channel Planning Best Practices. [Viitattu 19.4.2018] Saatavissa: [https://documentation.meraki.com/MR/WiFi Basics and Best Practices/Channel Planning Best Practices](https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Channel_Planning_Best_Practices)
12. Emerson Process Management. Introduction to HART (2002). [Viitattu 16.4.2018] Saatavissa: <http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/Eng%20Sch%20-%20Buses%20201.pdf>
13. Wordpress, Learnprotocols. Hart protocol, Rishabh Mishra (28.7.2011). [Viitattu 16.4.2018] Saatavissa: <https://learnprotocols.wordpress.com/2011/07/28/hart-protocol/>
14. STMicroelectronics. Who we are (2018). [Viitattu 13.4.2018] Saatavissa: http://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html
15. STMicroelectronics. Wi-Fi. [Viitattu 13.4.2018] Saatavissa: <http://www.st.com/en/wireless-connectivity/wifi.html?querycriteria=productId=SC1930>
16. STMicroelectronics. SPWF01SA. [Viitattu 13.4.2018] Saatavissa: http://www.st.com/content/st_com/en/products/wireless-connectivity/wifi/spwf01sa.html
17. RSComponents. STMicroelectronics Serial 2.4GHz WiFi Module. [Viitattu 19.4.2018] Saatavissa: <https://uk.rs-online.com/web/p/radio-frequency-development-kits/8787358/>
18. STMicroelectronics. SPWF04SA. [Viitattu 13.4.2018] Saatavissa: http://www.st.com/content/st_com/en/products/wireless-connectivity/wifi/spwf04sa.html
19. DigiKey. STMicroelectronics SPWF04SA. [Viitattu 19.4.2018] Saatavissa: <https://www.digikey.com/product-detail/en/stmicroelectronics/SPWF04SA/497-17232-ND/7071824>

20. STMicroelectronics. Microcontrollers. [Viitattu 13.4.2018] Saatavissa: <http://www.st.com/en/microcontrollers.html>
21. STMicroelectronics. Evaluation tools. [Viitattu 16.4.2018] Saatavissa: <http://www.st.com/en/evaluation-tools.html>
22. STMicroelectronics. NUCLEO-L476RG. [Viitattu 16.4.2018] Saatavissa: <http://www.st.com/en/evaluation-tools/nucleo-l476rg.html>
23. STMicroelectronics. X-NUCLEO-IDW01M1. [Viitattu 16.4.2018] Saatavissa: <http://www.st.com/en/ecosystems/x-nucleo-idw01m1.html>